AMENDMENTS TO THE SPECIFICATION:

Page 1, line 1, before BACKGROUND OF THE INVENTION, please insert as follows:

This application claims priority to United Kingdom Application No. 0024311.3 filed October 4, 2000, the entire contents of which are incorporated herein by reference.

Please amend the paragraph beginning at page 3, line 6, as follows:

The invention provides a new data processing instruction within a data processing system that may serve to unpack data values held within a data word and also perform a single-instruction-multiple-data type arithmetic operation upon the unpacked data values. The invention recognises that ~~by~~ unpacking non-adjacent data values within a data word may be implemented with considerably less additional overhead than conventional unpacking instructions which unpack adjacent data values. In particular, the need for additional data pathways that can diverge the bit positions of previously adjacent data values may be avoided. Instead, for example, already present masking and word shifting circuitry may be utilised. Furthermore, the simplification of the unpacking function allows the possibility for a single ~~instructions~~ instruction to also provide an arithmetic operation upon the operands without introducing processing cycle constraint problems.

Please amend the paragraph beginning at page 5, line 11, as follows:

Figure 5 schematically illustrates data path of a data processing system well suited for executing the data processing instructions of Figures 3 and 4. [[Please indent the paragraph]].

Please amend the paragraph beginning at page 5, line 16, as follows:

DETAILED DESCRIPTION ~~OF THE PREFERRED EMBODIMENTS~~

Please amend the paragraph beginning at page 6, line 1, as follows:

When the multibit portions have been selected, each is promoted in length from 8 bits to 16 bits using either zero or sign extension. The shaded portions of the promoted data word P shown in Figure 1 indicate these extension portions.

Please amend the paragraph beginning at page 6, line 20, as follows:

Figure 2 illustrates an example data path 2 of a data processing system that may be used to implement the instruction of Figure 1. A register bank 4 holds 32-bit data words to be manipulated. Both the input operand data words stored in Rm and Rn are read from this register bank and the ~~result~~ resulting data word is written back to register Rd in the register bank 4. The data path 2 includes a shifting circuit 6 and an adder circuit 8. The many other data processing instructions provided by the system utilise this shifting circuit 6 and adder circuit 8 in various different ways. Such a data path 2 is carefully designed so that the time taken for a data value to propagate through the shifting circuit 6 and the adder circuit 8 is well matched to the data processing cycle time. Efficient use of the hardware resources of the data path 2 is made in systems in which those resources are active for a high proportion of every data word propagating through the data path 2. A sign/zero extending and masking circuit 10 is provided in parallel with a lower portion of the shifting circuit 6. A ~~multiplex~~ multiplexer 12 is able to select either the output of the full shifting circuit 6 or the output of the sign/zero extending and masking circuit 10 as one of the inputs to the adder circuit 8. The other input to the adder circuit 8 is the input operand data word of Rn.

Please amend the paragraph beginning at page 8, line 21, as follows:

In operation, the unshifted input operand data word of Rn passes directly from the register bank 16 to the selecting and combining logic 22. In the case of instruction of Figure 3, the most significant 16 bits of the value of Rn are selected and form the corresponding bits

within the output data word Rd. In the case of the instruction of Figure 4 it is the least significant 16 bits of the input operand data word of Rn that are selected and passed to form the least significant bits of the output data word Rd. The input operand data word of Rm passes through the full shifting circuit 18. In the case of the instruction of Figure 3, an arithmetic right shift of k bit positions ~~in~~ is applied and then the least significant 16 bits from the output of the shifting circuit 18 are selected by the selecting and combining circuit 22 to form the least significant 16 bits of the output data word of Rd. In the case of the instruction of Figure 4, the shifting circuit 18 provides a left logical shift of k bit positions and supplies the result to the selecting and combining circuit 22. The selecting and combining circuit 22 selects the most significant 16 bits of the output of the shifting circuit 18 and uses these to form the most significant 16 bits of the output data word of Rd.